Problem 1
Prove that a Turing machine can decide the following language:

\[ L = \{ \langle F, w \rangle \mid F \text{ is a finite state automaton that does not accept input } w \} \]

Problem 2
Consider the following language, operating over \( \Sigma = \{a, b\} \):

\[ L = \{ w \mid w \text{ does not contain the substring } aba \text{ and has either one or two } b's \} \]

Write an FSA that decides \( L \). Please submit your solution to the Gradescope autograder as P2.txt.

Problem 3
Consider the following language, operating over \( \Sigma = \{a, b\} \):

\[ L = \{ w \mid w \text{ contains an even number of } a's \text{ and an odd number of } b's \text{ and does not contain the substring } ab \} \]

Write an FSA that decides \( L \). Write a regular expression that decides \( L \). Please submit your solution to the Gradescope autograder as P3.txt.
FSA Syntax

1 NAME, STATE, SIGMA, START, ACCEPT, and REGEX are all one line each, are each followed by a colon (:), and must all be present.

2 States can be a sequence of printable ASCII characters that do NOT contain whitespace or quotes.
   • q0, q1, q3, q1# are all fine
   • q_Looking_for_a_one is legal, but frowned upon
   • “q state here” is illegal and will confuse the FSA

3 Elements of Sigma can be a sequence of printable ASCII characters that are NOT whitespace or quotes. For clarity, do NOT reuse the names for states.
   • 0, 1, B, 1x, 0x, X, Y are all fine
   • “Crossed one out”, Ø, I are all illegal and will confuse the FSA

4 You are only required to write a regular expression for Problem 3. You may leave this field blank for Problem 2.

5 You do not need to worry about blank tape cells since the FSA will halt when it reaches the end of the input.

6 Recall delta is $Q \times \Sigma \rightarrow Q$

7 Each line has one transition. Each part of the transition is separated by whitespace.

8 You may list multiple accept states. The FSA will only accept an input string if it is in an accept state when it reaches the end of the input.

9 Any transitions not written in Delta will be treated as an implicit rejection of the input string.

10 Delta must end with END on a single line.

11 Anything after END will not be read by the simulator, so feel free to explain the machine and its states.

12 You may include comments in the code. A comment is on a line of its own starting with a semicolon.
NAME: Prof Monroe
STATE: q0 q1
SIGMA: a b
START: q0
ACCEPT: q1
REGEX: a[ab]*
DELTA:
q0 a q1
q1 a q1
q1 b q1
END

Comment however you like down here.
This FSA will accept any string beginning with an ‘a’